

**WEB-BASED IMAGING SERVICE PROVIDING THE ABILITY TO SPECIFY A
CHARGE-BACK ACCOUNT**

Shell S. Simpson
5196 North Maidstone Way
Boise ID 83713
Citizenship: U.S.A.

Ward S. Foster
3103 Hillway Drive
Boise ID 83702
Citizenship: U.S.A.

Kris R. Livingston
11232 West Hickory Bark Drive
Boise ID 83713
Citizenship: U.S.A.

RELATED APPLICATION

The present application is related to co-pending and commonly assigned U.S. Patent Application Serial Number 09/712,336 entitled "SYSTEM AND METHOD FOR PROCESSING DATA IN A DISTRIBUTED ENVIRONMENT," filed November 13, 2000; co-pending and commonly assigned U.S. Patent Application Serial Number 09/874,184
5 entitled "SYSTEM AND METHOD FOR PRINTING FROM A WEB APPLICATION," filed June 4, 2001; co-pending and commonly assigned U.S. Patent Application Serial Number 09/874,427 entitled "DYNAMIC PRODUCTION DEVICE REPRESENTATION IN A DISTRIBUTED ENVIRONMENT," filed June 4, 2001; and co-pending and commonly
10 assigned U.S. Patent Application Serial Number 09/924,058 entitled "SYSTEM AND METHOD AND PROGRAM PRODUCT FOR MULTIUSER PROFILE OPERATIONS AND GROUP COMPOSITION STORE" filed August 8, 2001, the disclosures of which are all hereby incorporated herein by reference.

TECHNICAL FIELD

The present invention relates to web-based imaging services, and more particularly to web-based imaging services providing the ability to specify a charge-back account.

25032751.1

BACKGROUND

To collect charge-back information, current solutions require software to be installed on the client machine. It would be desirable in the art to provide a system and method of charge-back account accounting that could be remote and/or distributed, platform
5 independent, and easy to use.

25032751.1

SUMMARY OF THE INVENTION

The present invention is directed to a system and method which provides a user the ability to specify a charge-back account. From a client browser, the user accesses a destination service representing a production device, e.g., printer, which downloads web content into the browser and retrieves the user's imaging data, e.g. from user's identity or from a hard disk. The destination service displays in the browser a menu including production options to process the user's imaging data and a list of charge-back account IDs. In some embodiments, the display includes only charge-back IDs that are customized to the user. The user selects production options and can specify a charge-back account ID. In some embodiments the charge-back account ID is required for certain options or for the entire production job to proceed. In some embodiments a cost breakdown is calculated and/or displayed for the job.

In some embodiments, following specifying a charge-back account, a list of default production options associated with the specified charge-back account is displayed. In some variations, the account is not charged unless processing (e.g., printing) completes as expected.

Among other technical advantages, embodiments of the present invention provide a flexible means of doing charge-back accounting within web-based imaging architecture that is platform independent. Lists of charge-back accounts customized to the user and default processing options customized to individual charge-back accounts greatly streamline job order entry. Some embodiments use imaging information available through web-based imaging to analyze the resources required and used in the charge-back, although not all embodiments are restricted to web-based imaging. In some implementations the solution can be deployed entirely on the destination service, where it can be maintained effectively.

BRIEF DESCRIPTION OF THE DRAWING

FIGURE 1A is a simplified schematic diagram representing a logical overview of a typical web-based imaging system, in accordance with embodiments of the present invention;

FIGURE 1B is a simplified schematic diagram depicting various aspects of destination services, in accordance with embodiments of the present invention;

FIGURE 1C is a schematic diagram illustrating in more detail various aspects of the network of FIGURE 1A, in accordance with embodiments of the present invention;

FIGURE 2 is a simplified block diagram illustrating schematically a web-based imaging configuration providing the ability to specify a charge-back account, according to embodiments of the present invention;

FIGURE 3A is a flow diagram depicting operation of a web-based imaging service providing the ability to specify a charge-back account, according to embodiments of the present invention;

FIGURE 3B is a flow diagram depicting an alternative operation to that described in connection with FIGURE 3A;

FIGURE 4 is a flow diagram further depicting the continuation of operation following the flow diagram of the web-based imaging service of FIGURE 3A;

FIGURE 5A is a schematic diagram depicting a client-server network system in accordance with embodiments of the present invention; and

FIGURE 5B is a schematic diagram depicting a variation of the client-server network system of FIGURE 5A, which is tailored to faster data rates or limited client machine storage capacity

GLOSSARY OF TERMS AND ACRONYMS

The following terms and acronyms are used throughout the Detailed Description:

“API”. An application programming interface (API) is a library of programmatic methods provided by a system of some kind (an example is a web-based imaging system, as in the present invention) that enables client programs (web application content operating within the browser is one example) to interact with that system. One method of creating an API is to create a library. For example, in JAVA™, a library (conventionally called a jar file) is created by defining a class or classes, compiling the class or classes, and grouping the class or classes into a library. For example, the following class could be created:

```
class BaseConversionAPI {static public String convertBaseToBase (String in
Number, int inBase, int outBase) { // Code for returning a string representing
inNumber converted to outBase}}
```

That class would then be compiled with the command:

```
java.exe BaseConversionAPI .java
```

NOTE: Programs are typically stored in text files, which are "compiled" in order to create "object files" which contain the executable (or interpretable) instructions. In this case, the program is contained in the file BaseConversionAPI.java. The act of compiling creates a file named "BaseConversionAPI.class" containing instructions for a specific computing architecture (in this case the JAVA™ Virtual Machine) corresponding to the program.

Next in this example, a Jar file would be created:

```
jar.exe cvf BaseConversionAPI.tar BaseConversionAPI.class
```

This command creates a "library" file containing the BaseConversionAPI class. This last step is not absolutely required. In some instances, API's are provided as simply files containing executable instructions (such as the BaseConversionAPI.class file).

References regarding the creation of API's:

<http://www.library.yale.edu/orbis2/public/activity/AP.html>

Note that the API's to network services (graphic store, composition store, and user profile store, all to be discussed below) would be created to be accessible through a remote invocation technology such as CORBA, JAVA™-RMI, DCOM™, RPC, or SOAP. A wide variety of references are available that describe how API's can be created to be accessible through a remote invocation technology, such as one of the technologies noted above.

"Client-Server". A model of interaction in a distributed system in which a program at one site sends a request to a program at another site and waits for a response. The requesting program is called the "client," and the program which responds to the request is called the "server." In the context of the World Wide Web (discussed below), the client is a "Web browser" (or simply "browser") which runs on the computer of a user; the program which responds to browser requests by serving Web pages, or other types of Web content, is commonly referred to as a "Web server."

"Composition." Composition, also referred to as a "graphics composition," comprises a file with links to graphic data serviced as a single unit, i.e., a graphic. The file also usually includes information on the placement of those graphics on a sequence of canvases. It describes how to combine one or more graphics from one or more sources onto a sequence of canvasses, in a variety of different ways. The use of compositions allows multiple compositions to reference a graphic in a graphic store without having to duplicate the graphic.

"Composition store". Composition store refers to a service (ideally implemented as a network service) that stores and provides access to imaging composition(s) that can be accessed by the user or web services. In this context, providing "access" includes providing methods for building compositions, modifying compositions, and accessing them piecemeal. For example, a set of methods available for execution via the composition store might include

the methods Get a Composition, Create a Composition, Delete a Composition, and Modify a Composition.

“Content.” A set of executable instructions that is served by a server to a client and that is intended to be executed by the client so as to provide the client with certain
5 functionality. Web content refers to content that is meant to be executed by operation of a Web browser. Web content, therefore, may non-exhaustively include one or more of the following: HTML code, SGML code, XML code, XSL code, CSS code, JAVA™ applet, JavaScript™ and C-"Sharp" code.

“Exchange infrastructure.” An exchange infrastructure is a collection of services distributed throughout a network that store imaging data associated with a particular user through a user profile.

“Firewall.” A firewall filters out unwanted communication packets in one or more directions. By way of example, in one implementation of a firewall, requests from inside a firewall may be made to access data on the outside of the firewall, and responses to such
5 requests are typically permitted. Communications initiated from outside the firewall to devices inside of the firewall are typically not permitted. Generally, the firewall may be implemented by a firewall proxy server that allows devices inside the firewall to pass HTTP requests to web servers outside the firewall. Obviously, other protocols may be used to implement communication through the firewall.

20 “Generic access instructions.” A generic access instruction refers to an executable instruction that is intended to cause the executing device to generate generic access requests in order to access a set of target graphic data. These instructions call methods provided by, for example, an imaging extension, but are executing within a JVM/JAVA™ or similar environment (which the imaging extension is part of). Methods provided by the environment
25 in which the program is executed are typically called an “Application Programming Interface” (API). Note that a generic access instruction does not include the location of the target graphic data. Typically, the target graphic data is pre-selected (generally by a user)

and its location is determined from information that is maintained locally within the executing device.

For purposes of this application, the term “generic access instruction” refers to an executable instruction that is intended to cause the executing device to generate generic access requests in order to access a set of target data. A generic access instruction, however, does not include the location of the target data itself and neither does the generic access requests. Importantly, the target data is pre-selected (typically by a user) and its location is determined from information that is maintained locally within the executing computer or otherwise associated with the user. For this reason, the target data for a particular computer is said to be “associated” with that computer or more specifically with that user. Thus, for example, the target data that is associated with computer “A” is the data that computer “A” will access in response to a generic access instruction. The target data that is associated with computer “B” is the data that computer “B” will access in response to the identical generic access instruction.

Furthermore, in the case wherein the target data represents an image, that image is referred to herein as the “target image.” In this simplified example, it will be assumed that all generic access instructions specified by the system wide standard mentioned above are for accessing data that describes an image.

“Graphic data.” Graphic data refers to digital data capable of being represented as two dimensional graphics, such as a Portable Document Format (“PDF”) file or a Joint Photographic Experts Group (“JPEG”) file.

“Graphic store.” Graphic store refers to a network service or a storage device for storing graphics data that can be accessed by the user or other network services. The graphic store preferably accepts the graphic data in multiple standard file formats, and the graphic data is converted into these file formats when necessary depending on the implementation.

“Hyperlink.” A navigational link from one document to another, from one portion (or component) of a document to another, or to a Web resource, such as a JAVA™ applet.

Typically, a hyperlink is displayed as a highlighted word or phrase that can be selected by clicking on it using a mouse to jump to the associated document or document portion or to retrieve a particular resource.

5 "Hypertext System." A computer-based informational system in which documents (and possibly other types of data entities) are linked together via hyperlinks to form a user-navigable "web."

10 "HTML" (HyperText Markup Language). A standard coding convention and set of codes for attaching presentation and linking attributes to informational content within documents. (HTML 2.0 is currently the primary standard used for generating Web documents.) During a document authoring stage, the HTML codes (referred to as "tags") are embedded within the informational content of the document. When the Web document (or HTML document) is subsequently transferred from a Web server to a browser, the codes are interpreted by the browser and used to display the document. Additionally in specifying how the Web browser is to display the document, HTML tags can be used to create links to other Web documents (commonly referred to as "hyperlinks"). For more information on HTML, see for example Ian S. Graham, The HTML Source Book, John Wiley and Sons, Inc., 1995 (ISBN 0471-11894-4).

15 "HTTP" (HyperText Transport Protocol). The standard World Wide Web client-server protocol used for the exchange of information (such as HTML documents, and client requests for such documents) between a browser and a Web server. HTTP includes a number of different types of messages which can be sent from the client to the server to request different types of server actions. For example, a "GET" message, which has the format GET <URL>, causes the server to return the document or file located at the specified URL.

20 "Imaging composition." An imaging composition comprises links to imaging data serviced as a single unit.

“Imaging data.” Imaging data refers to digital data capable of being represented as two dimensional graphics, such as a Portable Document Format ("PDF") file or a Joint Photographic Experts Group ("JPEG") file.

5 “Imaging data store.” Imaging data store refers to a network service or a storage device for storing imaging data that can be accessed by the user or other network services. The imaging data store preferably accepts the imaging data in multiple standard file formats, and the imaging data is converted into these file formats when necessary depending on the implementation.

10 “Internet.” A collection of interconnected or disconnected networks (public and/or private) that are linked together by a set of standard protocols (such as TCP/IP and HTTP) to form a global, distributed network. (While this term is intended to refer to what is now commonly known as the Internet, it is also intended to encompass variations which may be made in the future, including changes and additions to existing standard protocols.)

15 “PDA” (Personal Digital Assistant). A small hand-held computer used, for example, to write notes, track appointments, send email and browse the web with generally with far less storage capacity than a desktop computer.

20 “Personal imaging repository.” A personal imaging repository is a conceptual term describing the exchange infrastructure used to exchange graphics composition and graphics data with web services. Users are associated with their graphics data through user profiles. It should be noted that the personal imaging repository 50 can represent any type or combination of data storage devices.

“URL” (Uniform Resource Locator). A unique address which fully specifies the location of a file or other resource on the Internet or a network. The general format of a URL is protocol: //machine address: port/path/filename.

25 “User Information.” User information is identification and security information used in accessing graphics composition(s) and graphics data associated with a particular user

profile. It is preferably accessed either directly or indirectly through methods provided by an extension component integrated into the web browser.

“User Interface.” The junction between a user and a computer program providing commands or menus through which a user communicates with a program. For example, in the client-server model defined above, the server usually generates and delivers to a client a user interface for communicating with a program operating on or controlled by the server device. Where the server is a web server, the user interface is a web page. The web page when displayed by the client device presents a user with controls for selecting options, issuing commands, and entering text. The controls displayed can take many forms. They may include push-buttons, radio buttons, text boxes, scroll bars, or pull-down menus accessible using a keyboard and/or a pointing device such as a mouse connected to a client device. In a non-graphical environment, the controls may include command lines allowing the user to enter textual commands.

“World Wide Web” (“Web”). Used herein to refer generally to both (i) a distributed collection of interlinked, user-viewable hypertext documents (commonly referred to as Web documents or Web pages) that are accessible via the Internet, and (ii) the client and server software components which provide user access to such documents using standardized Internet protocols. Currently, the primary standard protocol for allowing applications to locate and acquire Web documents is HTTP, and the Web pages are encoded using HTML. However, the terms “Web” and “World Wide Web” are intended to encompass WAP and WML for mobile phone web browsers, as well as other current and future markup languages and transport protocols which may be used in place of (or in addition to) HTML and HTTP.

“Web Site.” A computer system that serves informational content over a network using the standard protocols of the World Wide Web. Typically, a Web site corresponds to a particular Internet domain name, such as “HP.com,” and includes the content associated with a particular organization. As used herein, the term is generally intended to encompass both (i) the hardware/software server components that serve the informational content over the network, and (ii) the “back end” hardware/software components, including any non-standard

or specialized components, that interact with the server components to perform services for Web site users. Importantly, a Web Site can have additional functionality, for example, a Web site may have the ability to print documents, scan documents, etc.

“Web service.” A web service is intended to refer to a service that is provided (at least in part) by a web server. But a web service is a broader concept than a web server. In this regard, a “Web server” is a program that, using the client/server model and the World Wide Web's Hypertext Transfer Protocol (Hypertext Transfer Protocol), serves the files that form Web pages to Web users (whose computers contain HTTP clients that forward their requests). Every computer on the Internet that contains a Web site must have a Web server program. The most popular Web servers currently are Microsoft's Internet Information Server (Internet Information Server), which comes with the Windows NT server; Netscape FastTrack™ and Enterprise™ servers; and Apache, a Web server for UNIX-based operating systems. Other Web servers include Novell's Web Server for users of its NetWare™ operating system and IBM's family of Lotus Domino servers, primarily for IBM's OS/390™ and AS/400™ customers.

Web servers often come as part of a larger package of Internet- and intranet-related programs for serving e-mail, downloading requests for File Transfer Protocol files, and building and publishing Web pages. This larger package is referred to as the web service. Parameters for a Web server include how well it works with various operating systems and other servers, its ability to handle server-side programming, and publishing, search engines, and site building tools in the package.

DETAILED DESCRIPTION

FIGURE 1A is a simplified schematic diagram representing a logical overview of a typical web-based imaging system, in accordance with embodiments of the present invention. User 100 interfaces with client program 16, typically a web browser, which is logically connected through data path 51 with network 10. Also logically connected with network 10 through respective data paths 52-55 are for example among other entities source service 24, an imaging service 32, destination service 34, and imaging store 30 (for further description of

a source service, a destination service, and an imaging store see co-pending and commonly assigned U.S. Patent Applications Serial Numbers 09/712336, 09/874184, 09/874427, and 09/924,058, cited above, the disclosures of which have been incorporated herein by reference). Network 10 can be any of a variety of network types, including for example Internet, Intranet, and Ethernet, and the transmission medium of network 10 and data paths 51-55 can include electrically conductive cable, optical fiber, semiconductor, wireless, or any combinations of these. Data paths 51-55 need not be physical links but can represent data flows through any media. In general a web-based imaging system can include multiple client programs 16, source services 24, destination services 34, and imaging stores 30 each interconnected with a network 10 and having a unique network address, typically represented by a Uniform Resource Locator (URL). Imaging service 32 is a logical entity providing client program 16 access to multiple destination services 34 by accessing and downloading interfaces, typically web pages conventionally generated using HyperText Markup Language (HTML) coding. Web documents are conventionally located and acquired throughout network 10 using HyperText Transfer Protocol (HTTP).

FIGURE 1B is a simplified schematic diagram depicting various aspects of destination services 34, in accordance with embodiments of the present invention. A destination service 34 typically is a destination web service that represents one or more production devices 152, 154 on network 10. Production devices 152, 154 include printers; paper handling accessories such as binders, sorters, or folders; e-mail clients; facsimile devices; web servers; and data storage devices. Production devices are not, however, limited to those above, but may include any devices capable of electronically or physically saving, displaying, formatting, or transferring a target image. Some production devices perform a single type of service, for example printing, whereas other production devices perform multiple services. A self-representing production device 152 contains an embedded destination service 34, which represents production device 152 on network 10, allowing production device 152 to be connected directly to network 10 and accessed directly by client program 16. Client program 16 is typically a web browser that runs in a client machine 12, commonly a desktop or laptop and potentially a handheld computer or personal digital assistant (PDA). On the other hand, a production device 154 such as a conventional printer is

incapable of self-representation and consequently must be connected to and controlled by an external destination service 34 running on an intermediate device such as a desktop computer or a print server machine.

In some embodiments of the present invention, source service 24 generates a set of data representing a printable version of a target image, which includes a controlled symbol referring to a predetermined symbol set. Only when the printable version of the target image is accessed by an appropriate destination service that contains the predetermined symbol set, for example appropriate destination service 35, can the controlled symbol in the target image be produced or displayed. Any other destination service 34 not containing the predetermined symbol set, including for example destination services accessible through imaging service 32, can print or display at most only a proxy symbol in place of the controlled symbol, when printing or displaying the image. The data representing the printable version of the target image are referenced by a composition stored in imaging store 30, as described in more detail below.

FIGURE 1C is a schematic diagram illustrating in more detail various aspects of network 10 of FIGURE 1A, in accordance with embodiments of the present invention. Client program 16 running on operating system 14 in client machine 12 is logically interconnected through data path 52 of network 10 with source service 24, typically a source web service that runs on server machine 22 and generates interfaces, typically web content 20. When client program 16 browses to source service 24, web content 20, usually including executable content, is downloaded into the browser window of client program 16. Executable content 20 accesses imaging store 30 via application programming interfaces (APIs) contained in a modified imaging extension 18 of client program 16, for example through data paths 55 and 56. For further description of imaging extensions containing APIs see co-pending and commonly assigned U.S. Patent Applications Serial Numbers 09/874184 and 09/924,058, cited above, the disclosures of which have been incorporated herein by reference. Modified imaging extension 18, described in more detail below, can be accessed by, for example, JAVA™ applets for accessing imaging store 30, although other web programming technologies can be used.

In some embodiments of the present invention, a preview version of the printable version of the target image is incorporated into web content 20 of accessed destination services 34, 35, including destination services 34, 35 accessed through imaging service 32 which provides links to source and destination services, and is previewed to user 100 through client program 16 in the context of the capabilities of accessed destination services 34, 35. When user 100 selects the “print now” option, the entire production process is controlled indirectly by user 100 through client program 16.

FIGURE 2 is a simplified block diagram illustrating schematically a web-based imaging configuration providing the ability to specify a charge-back account, according to embodiments of the present invention. User’s browser 16 accesses a destination service, for example destination service 34, representing production device 210, for example a printer. In the present example destination service 34 representing production device 210 is shown externally linked with production device 210. In some embodiments a destination service, such as destination service 218, is embedded in and exclusively represents a production device, for example printer 216. Destination service 34, for example a web service, downloads web content 20 into browser 16. Through imaging extension 18, web content 20 may access and retrieve image data referenced from user’s personal imaging repository 30. Among other functions, web content 20 provides an ability for the user to specify from items in a menu of production options, for example choices menu 202, containing various options for a print job to print the image data depending on the capabilities of printer 210, which can include but are not limited to some or all of a number of copies, paper source, duplex printing, color printing, collating, and job retention. Unit costs for the various production options, for example paper source, duplex printing, and color printing, are provided to destination service 34 by way of a schedule of services associated with printer 210.

Another option that can be included in choices menu 202 is a charge-back account. Destination service 34 representing printer 210, in order to provide a list of available charge-back accounts, will have communicated with an account database 212 associated with destination service 34 to obtain the list of accounts, which are provided to the user, for example, in a drop-down formatted accounts list 204 selectable from choices menu 202.

From accounts list 204 the user can choose an account, for example HP account XXX, for the current print job. In some versions only one account ID is assigned to each customer, for example account HP, whereas in other versions multiple account IDs, for example HP XXX, ..., HP XYZ, are assigned at the level of departments, types of print jobs, or even at the level of a single print job for a specific customer. In some implementations, a user can input for example the name, first portion of the name, or an abbreviated code for a customer to view a display of the account ID or list of all account IDs for that specific customer only. The account identification (ID) is included in the print job (or print job initiation) that is transmitted back to printer 210 through destination service 34. The print job variables are combined with the schedule of services and the account ID to generate an account billing for the print job in an account collection service 214 associated with account database 212.

FIGURE 3A is a flow diagram depicting operation of a web-based imaging service providing the ability to specify a charge-back account, according to embodiments of the present invention. At block 302 user's browser 16 accesses destination service 34 representing printer 210. At block 303 destination service 34 downloads web content 20 into user's browser 16. At block 304 web content 20 may access and retrieve image data from user's personal imaging repository 30 through imaging extension 18. At block 306 web content 20 displays in the window of user's browser 16 choices menu 202, which offers to the user various print job production options and the opportunity to supply an account ID. In some implementations, the list of account IDs 204 can be customized by destination service 34 for a particular user, as illustrated at block 305. When a particular user's browser 16 accesses a destination service 34, the destination service recognizes the particular user, for example using cookies, and web content 20 displays at block 306 only the account IDs that are relevant to and associated with that user. At block 307 the user proceeds to choose production options and to specify an account ID.

In some implementations, as illustrated at block 308, the user is required to specify an account ID before proceeding with the print job. In a variation illustrated at block 309, the user is prohibited selectively at block 310 from proceeding with specific production options only without specifying an account ID. At block 311 a cost breakdown for the print job is

calculated dynamically by destination service 34 in association with account database 212 and account collection service 214 as the user selects options. In some variations, some functions, for example color and collating, require charge-back information to be supplied, while others, for example job retention, do not. In some embodiments unit costs, discounts, and surcharges can be allocated on both an account ID basis and on a production option basis. These are factored into the dynamically displayed cost breakdown and also into final billing. Destination service 34 in association with account database 212 and account collection service 214 have the capability to capture, store, process, and output complex multidimensional matrices of costs associated with accounts and production options, and to display and charge-back the results. As illustrated at block 313, in some implementations this cost breakdown is displayed dynamically as the user selects options. The user then has the opportunity, for example as illustrated at block 316, to modify or delete selected options, or to discontinue the entire print job as depicted at block 317.

In some embodiments, specifying an account ID also specifies at block 312 a set of default production options associated with that account ID, which are displayed at block 313 in choices menu 202, and which the user has the opportunity to modify or override, for example as illustrated at block 316. In another variation, the job information available through web-based imaging or other means of accessing job specific information prior to processing the job is used to analyze the job, as illustrated at block 314, and to report paper, toner, or other resources required to process the job with the charge-back information at block 315.

FIGURE 4 is a flow diagram further depicting the continuation of operation following the flow diagram of the web-based imaging service of FIGURE 3A. Starting at block A (also shown in FIGURE 3A), web content 20 transmits the print job (or print job initiation) including account ID at block 402 to printer 210 via destination service 34, where the print job is executed as indicated at block 403, according to the user's selected production options, and the job is charged back to the account ID at block 405 as specified at block 307 of FIGURE 3A. In some implementations, the account is not charged unless processing (i.e. printing) completes as expected, as illustrated by the flow loop containing continuation block

407 around completion block 404. At block 406 an account billing and report for the print job can be generated by destination service 34, in some variations in association with account database 212 and account collection service 214.

Importantly, web based imaging in general allows a user to work with imaging information in a much more flexible way. Prior art alternatives include walking up to a printer, manually inputting an account ID, which the user consequently needs to know. In the present embodiments, account database 212 knows about the account IDs and provides that information to destination service 34 representing printer 210. Destination service 34 customizes web content 20 to include only what is relevant to the individual user. For example the user views a customized limited drop-down list of account IDs from which the user can specify a charge-back account. In some embodiments the user can specify an individual account ID that automatically specifies default production options, and the user has nothing further to do from that time on. Thus, it becomes more appealing to use web based imaging to make, for example, five copies of a particular document, as opposed to using a photocopier, where the user has to enter multiple codes manually.

Among other technical advantages, embodiments of the present invention use imaging information available through web-based imaging or other means of accessing job data prior to the processing of the job to analyze resources required and used in the charge-back. The embodiments provide a platform-independent means of doing charge-back accounting within a web-based imaging architecture. In some embodiments, the solution can be deployed entirely on a printer or on a destination service representing a printer, where it can be maintained effectively.

The principles of the present invention can additionally be applied to non-web-based imaging methods of retrieving image data for a processing job. FIGURE 3B is a flow diagram depicting an alternative operation to that described above in connection with block 304 of FIGURE 3A. In the alternative method of FIGURE 3B, user's browser 16 communicates for example with a local hard disk 19 (see FIGURE 2). Optionally, any type of local non-volatile large-scale digital storage device can substitute for local hard disk 19.

As described in connection with FIGURE 3A, at block 302 user's browser 16 accesses destination service 34 representing printer 210. At block 303 destination service 34 downloads web content 20 into user's browser 16.

5 Instead of web content 20 accessing and retrieving image data from user's personal imaging repository 30 through imaging extension 18 at block 304, at block 324 of FIGURE 3B the user selects a document stored on local hard disk 19. At block 325 the selected document is uploaded from local hard disk 19 to destination web service 34 representing printer 210. At block 326 destination web service 34 provides web content 20 to browser 16 that displays a preview image representing the document retrieved from local hard disk 19 at block 324. After this, operation proceeds as in FIGURE 3A to a display of choices menu 202 at block 306 or alternatively to optionally customizing list of account IDs at block 305 of FIGURE 3A, and thereafter follows sequentially the remaining operations depicted in FIGURES 3A and 4.

10 Broadly stated, the present invention is directed to an improved system and method for printing from a web application. The system and method provide printing from a web application that is independent of the configuration of the operating system. In addition, since the print destination server can return with specific print content that relates to a selected device, the present invention allows a preview of the print job in the context of the devices and/or services offered by the print destination server.

15 The system and method provide printing from a web application that is independent of the configuration of the operating system. In addition, since the print destination server can return with specific print content that relates to a selected device, the present invention allows a preview of the print job in the context of the devices and/or services offered by the print destination server.

20 The system and method provide printing from a web application that is independent of the configuration of the operating system. In addition, since the print destination server can return with specific print content that relates to a selected device, the present invention allows a preview of the print job in the context of the devices and/or services offered by the print destination server.

25 FIGURE 5A is a schematic diagram depicting client-server network system 10 in accordance with embodiments of the present invention. Client machine 12 is connected to first server machine 514 and second server machine 516 via Internet 518. Client machine 12 includes client program (browser) 16 and preferably personal imaging repository 522.

Browser 16 further includes extension component (imaging extension) 18 that makes use of user information 526 in order to provide an interface between content executing in browser 16 and personal imaging repository 522. More specifically, user information 526 is used for associating accesses through extension component 18 with the appropriate user's personal imaging repository. It should be noted that the user profile can associate different users or groups with personal imaging repository 522. For example, the user profile can associate a single user with a particular personal imaging repository, but, at the same time, this user can also have multiple user profiles, resulting in multiple personal imaging repositories associated with a single user. Similarly, the user profile can associate a group having multiple users with a particular personal imaging repository. A personal imaging repository, in this scenario, can be used by a group having a common association, such as a group project. As described, the user profile can be defined with great discretion and flexibility, and the above implementations are contemplated and within the scope of the present invention.

Although the preceding description defines the user profile broadly, it should be understood that in the present embodiment each user has one personal imaging repository. A personal imaging repository will not typically be associated with groups—it will typically be associated only with individuals, but could optionally allow several individuals to use the same repository. This personal imaging repository is defined by all the information and services that are relevant to performing imaging operations for the particular user. The “root” of a user's personal imaging repository is one or more user profiles, which are associated with the user through one or more sets of user information. The present invention is directed to implementing the concept of allowing a user's information to follow him/her around, i.e., be accessible from a variety of different locations, both inside a firewall and outside of the firewall, as well as from a variety of different machines.

Imaging extension 18 is configured to respond to the execution of generic access instructions from web application content 528 by generating/mapping these generic instructions to corresponding imaging client-specific commands of imaging client 16. However, this will happen only if user information 526 (containing references to the user's

profiles) is available to imaging extension 18, to access the user's personal imaging repository 522.

Imaging extension 18 can be regarded and implemented as an application programming interface (API). The API used for imaging extension 18 is preferably structured in accordance with a system wide standard. The generic access instructions for example from web application content 528, when executed, can cause imaging extension API calls to be issued to the API in order to access the user's personal imaging repository 522 via imaging client-specific instructions. It will be recognized by those of ordinary skill in the art that there are other ways (both hardware and software) to implement this same functionality. Embodiments of the present invention are not limited to any one way. In essence, imaging extension 18 provides means for accessing user information 526 and for providing an opaque interface between web application content 528 executing in browser 16 and personal imaging repository 522 and other functionalities of imaging client 16. An example implementation of the imaging extension will be discussed in more detail below.

In operation, browser 16 initially accesses a web site and using appropriate request commands (HTTP for the current generation of browsers), downloads web application content 528 therefrom, which includes a set of executable instructions intended to be executed in browser 16 to provide browser 16 with predetermined functionality. These executable instructions comprise generic access instructions (see definition above), which are system wide instructions expressed in some language (i.e., JAVA™), that call the resources of an imaging extension API to access the user's personal imaging repository 522 to perform web imaging operations. Such generic access instructions can be, by way of example but not by way of limitation, JAVA™, JavaScript™, and C-sharp instructions. A system wide standard preferably manifested as an API or set of APIs typically specifies "generic access instructions," "generic access requests," and "target graphics."

A variety of functionality can be provided by web application content 528 including, for example, executable instructions for imaging client 16 to display target graphics, i.e., show available graphics on the accessed web site. Another web application content can

include executable instructions for displaying a print button, and if the print button is clicked, causing imaging client 16 to generate a print job that describes a graphic in the personal imaging repository 522 of the user and to transmit the print job, for example, to printer 542. A web application content can also provide a preview of the target graphic. Accordingly,
5 web application content 528 refers to a set of executable instructions that are downloaded into browser 16 to perform a service requested by the user.

Browser 16 executes web application content 528, whether it is HTML interpreted and/or executed by browser 16 into marks displayed on a user's display, or JAVA™ and JavaScript™ or some other appropriate language. As previously noted, web application content 528 contains executable instructions that use the API provided by imaging extension 18 to indirectly access the user's personal imaging repository 522. For example, the executable instructions of the web application content can obtain an opaque access to the information from the user's profile (in order to specify the user's personal imaging repository) by interacting with a user profile store service (not shown).

In the discussion herein, the term "opaque reference" is used. An "opaque reference" is a reference that does not expose information about an underlying resource. The possessor of an opaque reference is unable to determine anything about the resource from the opaque reference or to modify the opaque reference so as to alter which resource is being referenced. (In contrast, if a URL is provided, for example, "http://www.hp.com", it would be fairly
20 straightforward for the web application content to modify the URL to refer to a different resource, for example, "http://www.other.com".)

The executable instructions of web application content 528 perform this access to obtain an opaque reference to the user's composition store 546 and graphic store 548. The web application content can further use the API provided by imaging extension 18 to add a
25 new graphic to graphic store 548 via opaque reference.

Imaging extension 18 is configured to prevent web application content 528 (i.e., the executable instructions from web service 530), from directly accessing arbitrary services and the user's personal imaging repository 522. In essence, web application content 528 uses

imaging extension 18 as a gateway to access everything in the user's personal imaging repository 522, including the information in the user profile.

This restricted access imposed on web application content 528 can be implemented using a variety of methods. The designer can implement the API for imaging extension 18 such that the API only accepts references from web application content 528 that were previously provided thereto by imaging extension 18. In essence, web application content 528 is then unable to supply references arbitrarily when calling the API provided by imaging extension 18. Web application content 528 running on imaging client 16, in order to communicate with imaging client resources and with user's personal imaging repository 522, must first obtain opaque references using the API of imaging extension 18. For example, if web application content 528 wants to access graphic store 548, web application content 528 is required to call a method (provided by the API of the imaging extension 18) that provides an opaque reference to graphic store 548. This reference can then be used in subsequent calls by web application content 528 to the API of imaging extension 18.

One approach to accomplishing this restriction is to create a session. For example, an imaging extension API for a particular operation might comprise:

CreateParticularOperationSession() : returns SessionID

PerformOperation(Parameter, SessionID id) : returns Boolean (which indicate a result)

DeleteParticularOperationSession (SessionID)

Accordingly, web application content 528 is required to call the imaging extension API to first create a session by calling CreateParticularOperationSession, which returns a SessionID. This SessionID is subsequently used to refer to the particular session. Next, web application content 528 calls the PerformOperation in the imaging extension API with particular input and the SessionID. Web application content 528 can perform a variety of manipulations, but cannot directly access parameters and operations which are "associated" with the SessionID, because the association is accomplished in a way that is "opaque" to the

client. The imaging extension API and that API alone knows how to use the SessionID to determine/map to imaging client parameters. Often, the SessionID will be a reference such as a pointer to a data structure containing information relevant to the session. This data structure can contain parameters and other pertinent information. When web application content 528 has completed its operation, web application content 528 calls DeleteParticularOperationSession in the imaging extension API with the SessionID as a parameter. This instructs the imaging extension API to free whatever resources (such as memory) are associated with the session. Note that if web application content 528 changes the SessionID, that will not allow web application content 528 to obtain restricted parameters, but will only confuse imaging extension 18 with the changed previously unseen SessionID.

The API provided by imaging extension 18 is typically implemented as a library of methods that provide controlled access to an API provided by the network services participating in user's personal imaging repository 522. This API is implemented to invoke the API provided by the user profile store, composition store 546, and graphic store 548. The API provided by imaging extension 18 is generally not accessed through remote invocation technology, although remote invocation technology can be implemented to access the APIs provided by the network services participating in the user's personal imaging repository 522. The API provided by imaging extension 18 is not an exact replication of APIs provided by the user profile store, composition store, and graphic store, since this API provides controlled access to those network services through (among other techniques) opaque references.

From the above description, it can be seen that web application content 528 is prevented from using the API provided by imaging extension 18 to access arbitrary services. The key to this restriction is that web application content 528 cannot supply the addresses for these arbitrary services. Web application content 528 can only refer to services through opaque references provided by the imaging extension API (not exposing the actual reference/URL to web application content 528). For example, web application content 528 can use the API to obtain a list of opaque references to available compositions. This list of opaque references instead would map to the real references/URLs in imaging extension 18, alone. Thus, in subsequently referring to these compositions, web application content 528

cannot supply a URL (which might be one of its own creation), because that created URL cannot map within imaging extension 18 to real resources. Instead, web application content 528 is required to use only references provided to it by the API, which make sense only in the context of the current session with that API. This restriction can be relaxed in circumstances
5 where web application content 528 provides references to resources available from the same network service in which web application content 528 originated. This is permitted, because web application content 528 already has a measure of access to the web service from which it originated (either when originally generated or subsequently), thus not acquiring any special access not already available to web application content 528.

10 Browser 16 uses web application content 528 that is provided by web server 530. When the user selects "print" in the web application content, web application content 528 among other things directs browser 16 to the print destination. Although one client machine 12 and two server machines 514, 516 are shown as examples, a broader implementation can involve multiple server machines to which client machine 12 has access and can
15 communicate. For better readability, a single client machine, server, production device, e.g., printer, or application has been and will be referred to and shown herein. However, it should be understood by showing only one or by the use of "a" that what is meant is "one or more". For example, although a single printer has been and will be described and shown, this printer may actually be a plurality of printers forming a printing resource. In such a situation, it is
20 understood that the present inventive concepts apply.

First server machine 514 includes first server 534. When browser 16 is directed to first server 534 addressed by a unique Uniform Resource Locator ("URL"), first content 536 is served by the first server to browser 16. Each content 536 is preconfigured with specific instructions depending on the type of service the server machine represents. Similarly,
25 second server machine 516 includes a second server 538 with a second content 540. Generally, the contents 536, 540 are different, because the services and/or access to devices provided by the servers are different. In FIGURE 5A, first server machine 514 is connected to single printing device 542, whereas second server machine 516 serves multiple printing

devices 544. Consequently, first content 536 and second content 540 are different from one another, each including separate instructions to browser 16.

Although it is shown that the servers represent only printing devices in this example, the server can represent other services. For example, the server can be an auction web site, such as ebay.com, which makes an auction page for the user when a graphic file is printed to the web site, or a check writing service. In embodiments of the present invention, the user can "print" to any one of many services. As a result, the use of the word "print" is intended to have a broad definition, which can be applied to many available devices or services. Whatever the services and/or device the servers provide, the content can include the instructions needed for the configuration. It is advantageous that a personal imaging repository 522 be implemented according to the present invention, to store data that can be accessed by these servers.

In the present embodiment, personal imaging repository 522 includes composition store 546 for storing composition(s) of the imaging data that are serviced as a single unit and an graphic store 548, i.e., digital memory, for storing the imaging data. An imaging composition generally comprises links to the imaging data (also known as graphics), which can be located at another service or services. Accordingly, composition store 546 stores only the imaging compositions. Graphic store 548, on the other hand, is any imaging data store located on any computer that contains the graphics. More specifically, each web service can have its own graphic store 548 available to the public.

For example, at some earlier time a user can print an article from a web service site, resulting in an imaging composition being created and stored in the user's composition store 546. The imaging composition contains only the link to the graphic for this article stored for example on first web service site 514. Consequently, the graphic for the article is not in the graphic store 548 located on client machine 12. Rather, the graphic is stored in a graphic store 548 located on web service site 514. Users will have a graphic store 548 that belongs to their user identification, where they can store imaging data, which is graphic store 548 shown in client machine 12. As a result, the term "personal imaging repository" 522 is a conceptual

term for an exchange infrastructure between the imaging data and the available web services on Internet 518. Similarly, the term "web" denotes millions of distinct servers that comprise the web. However, the web does not actually do anything itself. In embodiments of the present invention, the servers serving composition store 546 and graphic store 548 are
5 physical implementations of the personal imaging repository as a concept.

It should be noted that personal imaging repository 522 can represent any type of data storage device. In fact, the data storage device of personal imaging repository 522 does not necessarily have to be located with client machine 12. Personal imaging repository 522 can be located, for example, on another machine or segmented and distributed among multiple machines, which client machine 12 can access through Internet 518. Although it is frequently
10 advantageous to include personal imaging repository 522 with client machine 12, this can change as data rates become faster and the popularity of personal digital assistant ("PDA") devices increases. These alternative implementations are considered to be within the scope of the present invention.

FIGURE 5B is a schematic diagram depicting a variation 560 of the client-server network system of FIGURE 5A, which is tailored to faster data rates or limited client machine storage capacity. In this implementation, multiple users 562, 564 utilize the same client machines 566 through Internet 568. In this implementation, client machines 566 can include client computers that have less storage memory, such as a Personal Digital Assistant
15 ("PDA") or a laptop. Because of limited storage memory, personal imaging repository 570 for storing user's data 572 is located on a separate computer 574, which can be a server computer or just a linked client machine 566. In this example, separate computer 574 is a server, which will be herein referred to as store server 574 to distinguish it from other servers for printing. Users 562, 564 are assigned distinct user profiles (not depicted explicitly in
20 Figure 5B) for accessing personal imaging repository 570 through store server 574. The user profile (or profiles) is part of the user's personal imaging repository. Users 562, 564 each have a distinct personal imaging repository, although only a single personal imaging repository 570 is depicted explicitly in FIGURE 5B. Although not shown, personal imaging repository 570 can similarly be implemented with a composition store and a graphic store,
25

where user's data 572 can be stored. User's data 572 is broadly interpreted to include one or more user profile store services, one or more composition store services and/or one or more graphic store services.

In operation, a first user 562 initially accesses system 560 with a login name and password. Once first user 562 has accessed system 560, the first user then also has access to the personal imaging repository 570 that is linked to this first user's login name. Similarly, if a second user 564 logs in with a login name and password, the second user has access to system 560, including the personal imaging repository 570 that belongs to this second user's login name. In this implementation, users can access system 560 and their personal imaging repository 570 from any computers that have a browser and Internet access. As a result of the flexibility of the Internet, it is possible for users to access system 560 and their personal imaging repository 570 using a standard PDA and/or wireless web phone.

Web application content 576 can be used by the users through a browser 578 that is located on client machine 566. Similar to the previous implementation, web application content 576 is provided through web server 580. Browser 578 also contains extension 584 for accessing user information 586, 587 that associate the different user profiles assigned to users 562, 564 with their respective personal imaging repositories. User information is different from a user profile. User information references one or more user profiles associated with a particular user. As shown, each user profile has its own user information. Alternatively, the user information can also contain information for two or more user profiles. These other variations are contemplated and are within the scope of the present invention.

Users can access a variety of servers on the Internet for the printing of the target data from web application content 576. In this example, there is first server 588 and second server 590. First server 588 provides first print content 592 representing single printing device 594, and second server 590 provides second print content 596 representing a plurality of printing devices 598.

Upon the user selecting PRINT or PRINT PREVIEW from web application content 576, web application content 576 first directs browser 578 to request a specific URL, which

references a web page located on web server 580. Shown as an example in FIGURE 5B, a user creates a document using web application content 576, and from web application content 576 the user can elect to PRINT the document (i.e., the target data).

In response to receiving the request for the web page specified by the aforementioned URL, web server 580 constructs an imaging data of the target data. An imaging data refers to the printed output of the target data, which does not necessarily look the same as the target data, depending on the behavior of web application content 576. This step of constructing an imaging data may not be necessary, and depends on the implementation and configuration of the print destination. For example, if sending a graphic file to the earlier example of an auction site for making an auction page with the graphic file, an imaging data might not have to be generated. Instead, print destination server 588, 590 can accept the target data without further modification for compatibility. However, since it is hard to foresee what type of graphic files the web site will accept, the exemplary method is implemented with the intermediate step of constructing imaging data to guarantee uniformity and compatibility. Formats for the imaging data include JPEG, Graphics Interchange Format ("GIF"), Portable Network Graphics Format, Tagged Image File Format ("TIFF"), PDF and Microsoft Windows bitmap format ("BMP").

After web server 580 constructs the imaging data for the target data, web application content 576 transfers the imaging data to personal imaging repository 570. It is then determined whether personal imaging repository 570 is located on client machine 566 or on store server 574. If personal imaging repository 570 is located on the client machine 566, the imaging data is saved to personal imaging repository 570 without further connection. If, however, personal image repository 570 is located on store server 574, client machine 566 will connect to store server 574. It is then determined whether the connection is successful before a timeout, and client machine 566 will keep trying to connect to store server 574 until a timeout or successful connection occurs. Once the connection with store server 574 is successful, client machine 566 transfers the imaging data to store server 574 for storage in personal imaging repository 570.

After the imaging data is stored in personal imaging repository 570, web application content 576 directs browser 578 to the server indicated by print destination 594, which will be referred to as print destination server 588. It is next determined whether print destination server 588 is available for printing. An error message is sent to browser 578 if the print destination server 588 is not available. If, on the other hand, print destination server 588 is available, it will respond to browser 578 by returning a print content 592, which will be displayed on browser 578 for user configuration. Print content 592 is generally a web page that is designed according to the services that this print destination provides. Furthermore, there are a number of ways to implement print content 592, depending on the services available. For example, print content 592 can be configured to display a list of imaging data stored in personal imaging repository 570. In this scenario, print content 592 accesses personal imaging repository 570 to obtain the list for display to the user. The above described variations are contemplated and should be considered within the scope of the present invention.

As an example, if print destination server 588, 590 represents multiple printing devices, a page of the print application content may contain all the printing devices 594, 598 that are available for user selection. From this page of the print content the user selects a printing device, and another page is returned to the user with the imaging data and the configurations that are available for this particular printing device. Through the print content, the user is able to print or print preview the imaging data according to the configurations of printing devices 598. In the auction site example, users can preview the auction page that they configured before posting onto the auction list. As shown, the print content can be returned with multiple pages depending on the need of the services provided by the print destination server.

Once the user-selected configuration is finalized, the user can then select to PRINT or PRINT PREVIEW from print content 592. Print content 592 accesses the imaging data from personal imaging repository 570, and transfers the imaging data with the specified user configuration to print destination server 588, through which the imaging data is printed or displayed according to the specified configurations including selected printing device 594.

At the end, the print content can return a status page to indicate successful output at the printing device.

Personal imaging repository 570 is an example of the notion of "identity." The user has a network "identity" through which he/she is represented. Personal imaging repository 570 contains information associated with the user's identity. The foregoing description addresses a computing environment in which imaging extension 584 is used to make user information available to web content 592, 596 downloaded into browser 578. Imaging extension 584 makes information associated with the user's identity available. The primary purpose of imaging extension 584 is to provide access to information that is identified by user specific information 586, 587. In essence, this is a client-side approach to identifying user information.

Although it is possible to extend browsers on Windows operating systems, it becomes harder to do this for newer devices like PDAs. Consequently, it is desirable to support the notion of identity without requiring the browser to be extended. Alternatively, a server-side approach to identifying user information is possible. This can be accomplished by moving the logic normally present in web content 592, 596 running within browser 578 into web server 588, 590. Rather than web content 592, 596 accessing services specific to the user, web server 588, 590 directly accesses services specific to the user. In other words, the identity technology is server side instead of client side.

When using server side identity technology, because browser 578 no longer provides information regarding a user's identity, an "authentication website" can be used to provide such information. In such an arrangement, web content 592, 596 redirects browser 578 to the authentication website, which determines the identity of the user and then redirects browser 578 back to the web content 592, 596 with the user identity, including the location of the user's profile. For this purpose, it is assumed that all web imaging destinations have information regarding the authentication server. Once the user's identity is determined (i.e., the location of the user's profile is known), then web imaging destination can interact directly with services specific to the user, without intervention of imaging extension 584.